THM - LazyAdmin Writeup

Zackary Jordan

01/26/2026

1 Target Information

- **Target IP Address:** 10.66.188.129

2 Reconnaissance

I used RustScan to identify open ports on the target system.

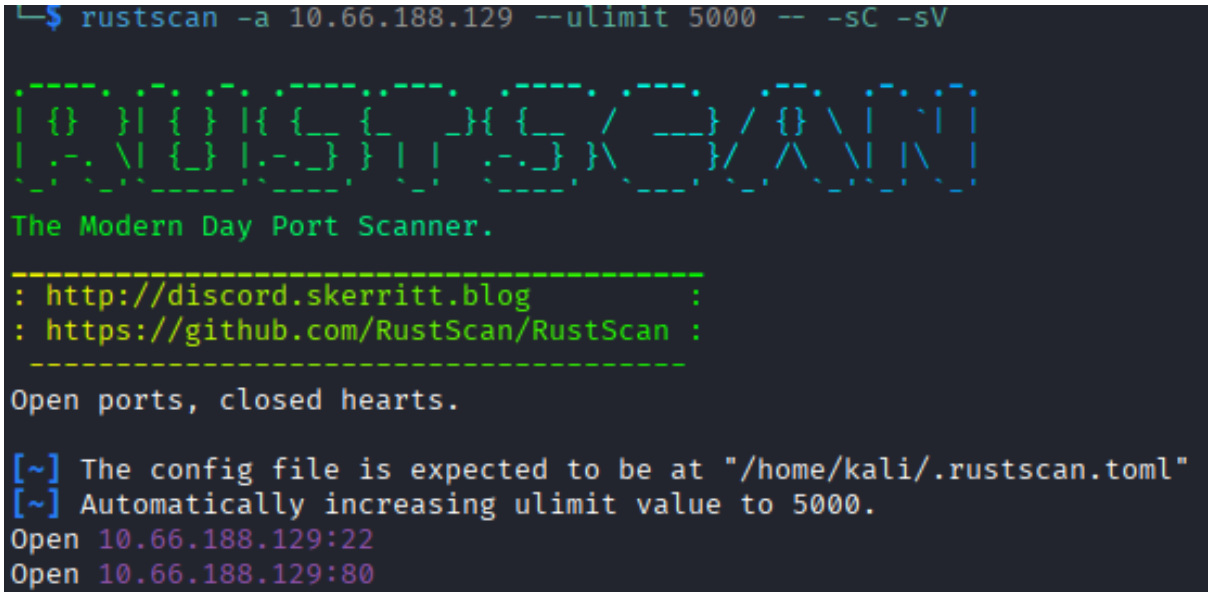

Figure 1: Rustscan Showing open ports on the target

3 Enumeration

3.1 Nmap Enumeration

Then I then used Nmap with service/version detection and default scripts to gain more information about the target.

```
nmap -sV -sC 10.66.188.129
```

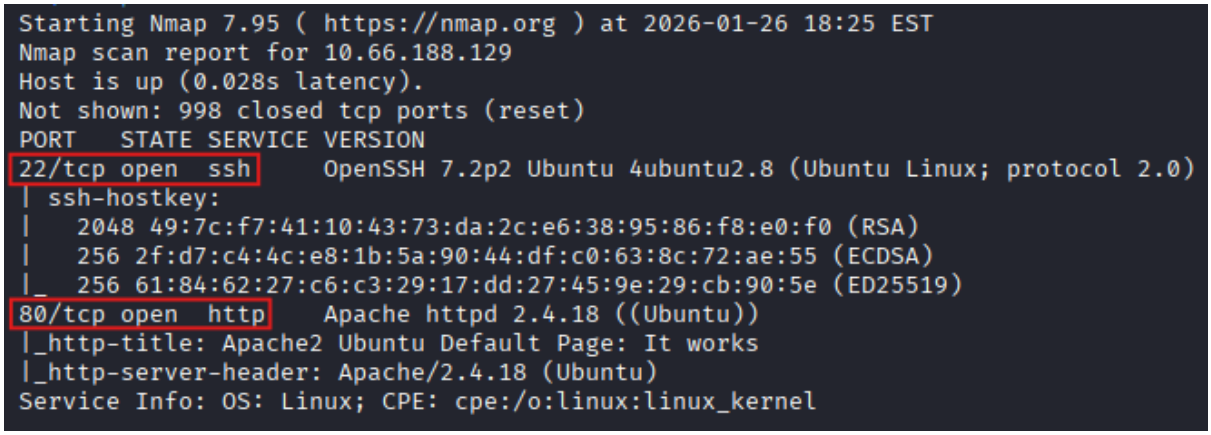

Figure 2: Nmap scan results showing SSH and HTTP services

3.2 Web Enumeration

From here, I accessed the main webpage, which revealed the Apache2 Ubuntu Default Page. Then, I performed directory enumeration using FFUF, which revealed multiple directories.

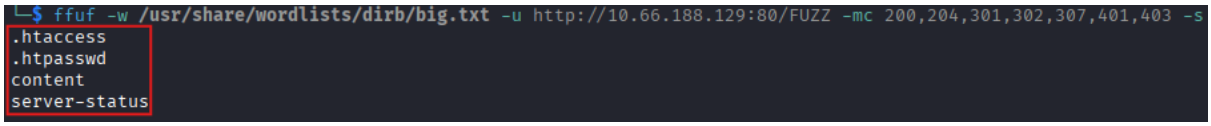

Figure 3: FFUF discovering common web directories

The only accessible and useful directory was `/content`, which indicated that the site was running the `SweetRice` website management system.

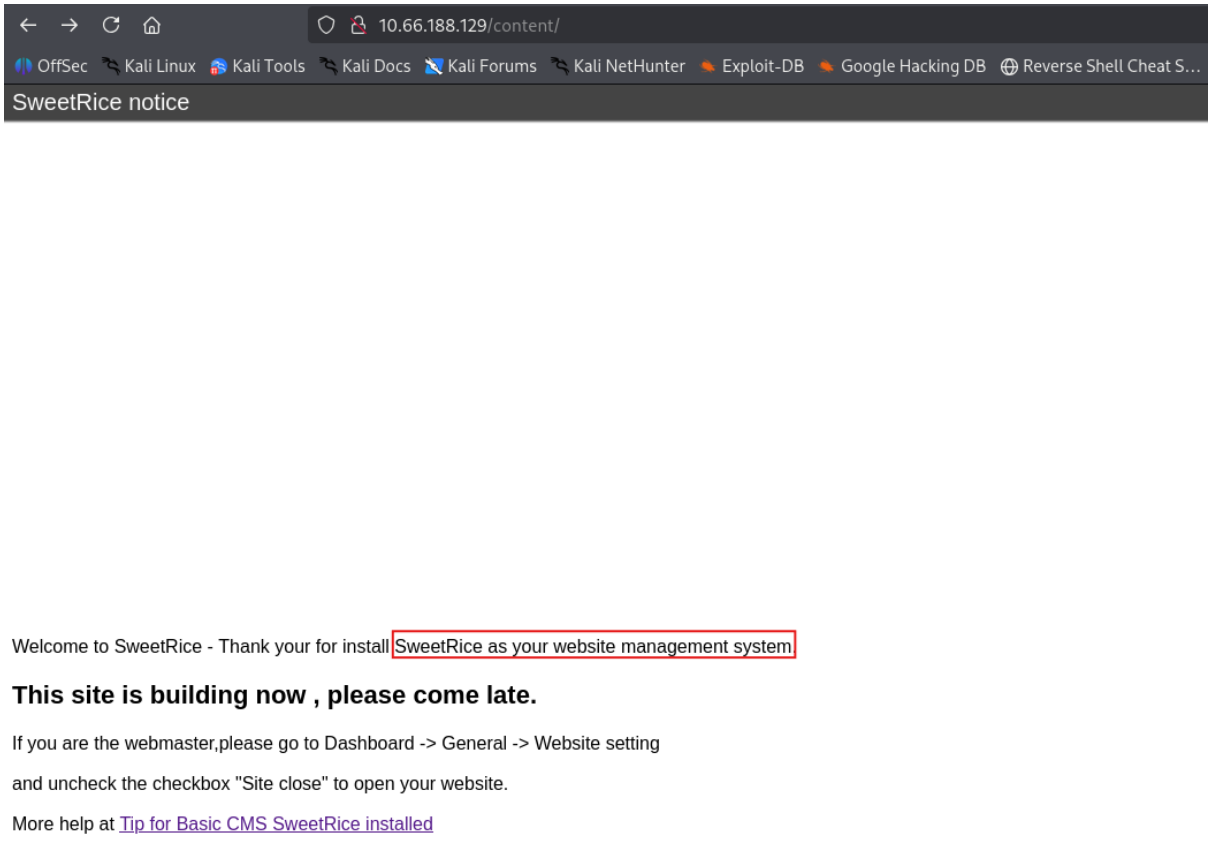

Figure 4: Content web directory showing SweetRice as website manager

After that, I used `SearchSploit` to enumerate known vulnerabilities for SweetRice, and found multiple potentially useful exploits.

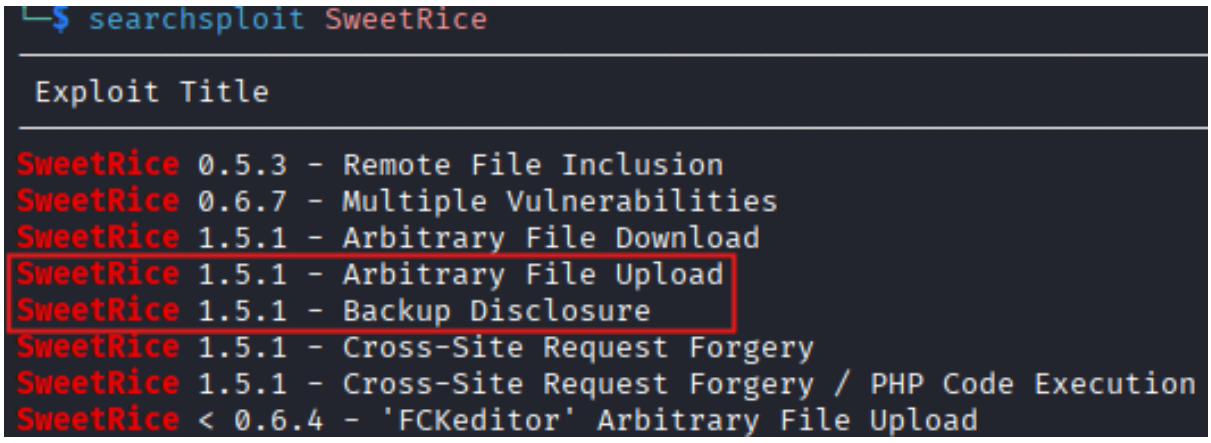

Figure 5: SearchSploit showed potential exploits

I selected `Arbitrary File Upload` as my main way to gain a shell and `Backup Disclosure` exploit for investigation, which required an `/inc` web directory containing a `mysql_backup`.

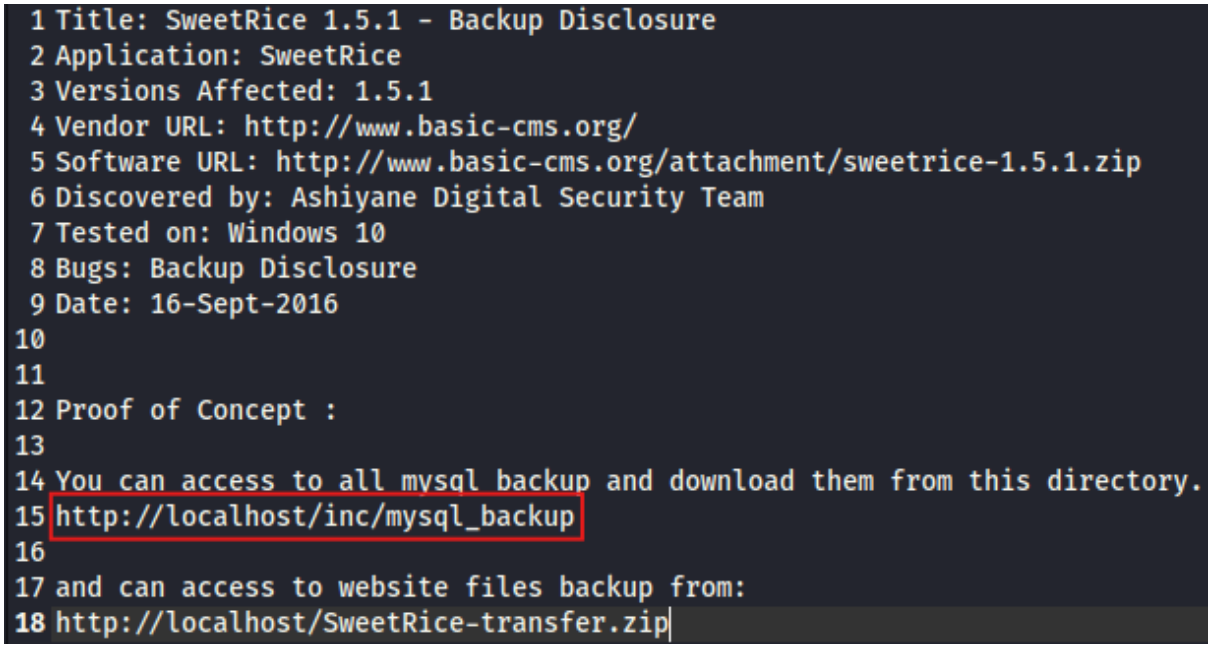

Figure 6: Backup Disclosure exploit needed info

I then ran an Additional FFUF enumeration against `/content`, which revealed an `/inc` directory and other endpoints.

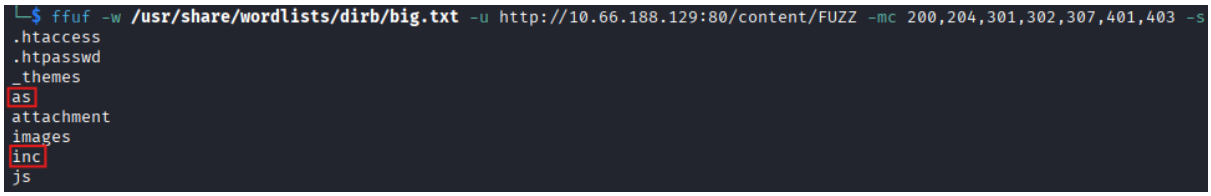

Figure 7: Additional FFUF scan of /content directory revealed additional sub-directories

I then found the `/as` directory was a login page and saved that for later. After, I then investigated the `/inc` directory and the saw the `mysql_backup`. Here, I downloaded the backup, revealing a potential username and hashed password that could be used for the login page I discovered earlier.

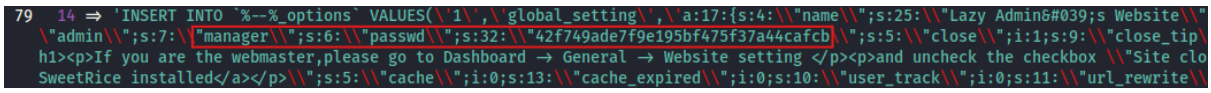

Figure 8: SQL Backup Dump revealed possible login info

From here I went to `CrackStation` and input the hash from the dump and was given a potential password for the username:`manager`.

Hash	Type	Result
42f749ade7f9e195bf475f37a44cafcb	md5	[redacted]

Figure 9: Crackstation successfully broke the hash for username:manager

4 Exploitation

Using the recovered credentials, I successfully authenticated to the SweetRice mainpage. Based on the `Arbitrary File Upload` vulnerability, I used the media center to upload a malicious file given A PHP reverse shell I selected from the system webshells and renamed for upload:

```
cp /usr/share/webshells/php/php-reverse-shell.php ICEb.phtml
```

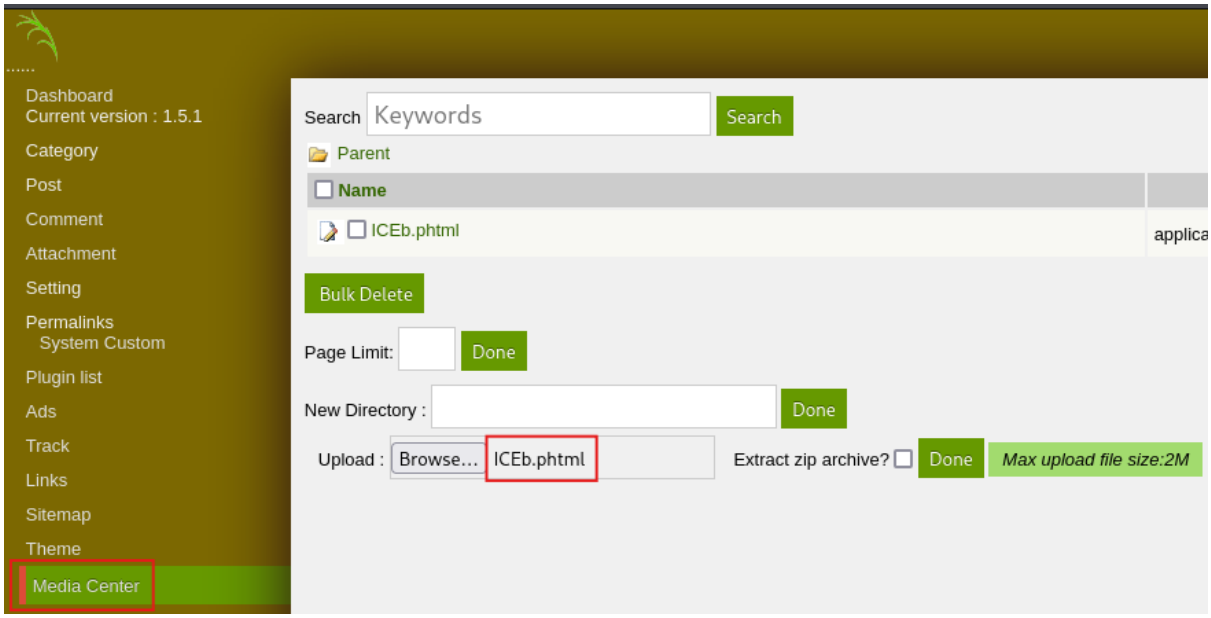

Figure 10: Successful upload of PHP reverse shell file

After uploading the file, I navigated to `/content/attachment`, located the uploaded payload, and executed it to obtain a reverse shell on the target.

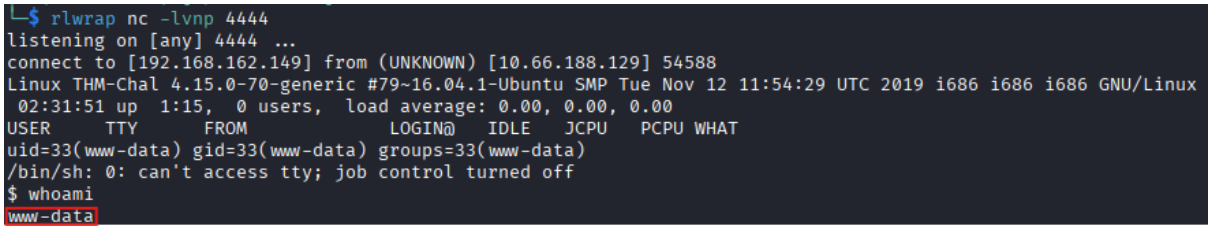

Figure 11: Successfully gained a shell using Arbitrary File Upload exploit

5 Post-Exploitation

After gaining initial access, I ran basic enumeration commands (e.g., `whoami`, `id`, `uname -a`) for proper identification.

I stabilized the shell using:

```
SHELL=/bin/bash script -q /dev/null
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Then restored job control with:

```
Ctrl-Z
stty raw -echo && fg
```

After stabilizing the shell, I navigated to `/home/itguy` and retrieved the `user.txt` flag.

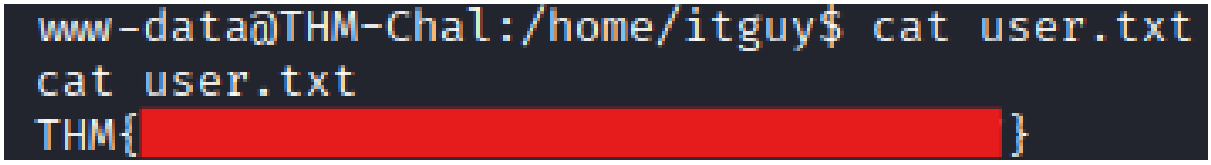

Figure 12: Acquired user.txt flag

6 Privilege Escalation

I first tried to enumerate the SUID binaries, but nothing useful was identified. I then ran `sudo -l` that revealed a Perl script that could be executed with elevated privileges.

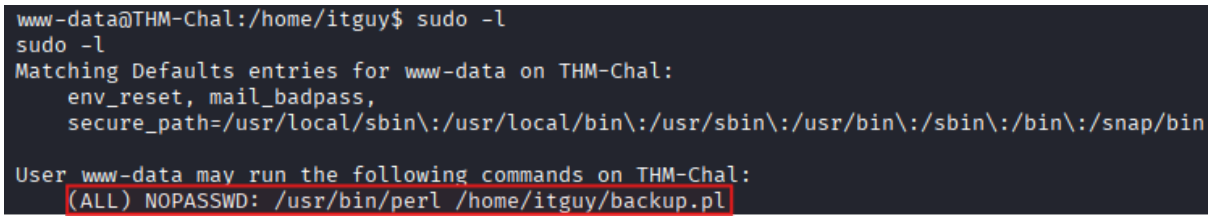

Figure 13: Possible Priv-Esc through found through sudo -l

I reviewed the script and confirmed it executed a Bash script using a system call.

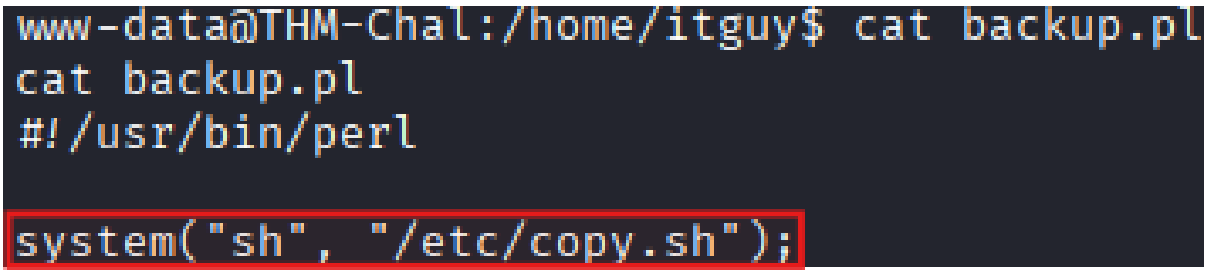

Figure 14: Investigated and found bash script being executed through system call

I then investigated `copy.sh` and found it already contained a reverse shell (some reason). Attempts to edit the file using standard editors failed, so I overwrote the file using `echo` and output redirection.

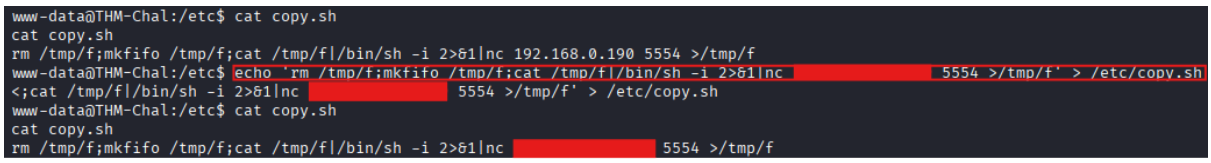

Figure 15: Echo write-over technique for attempt at priv-esc

After overwriting the script, I executed the sudo-allowed Perl script which resulted in a root shell.

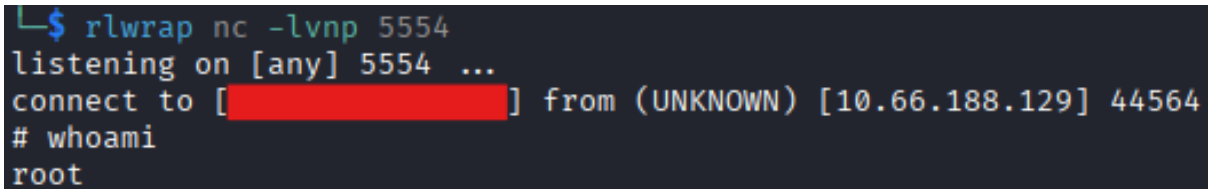

Figure 16: Successfully gained a root shell

Finally, I navigated to `/root` and retrieved the `root.txt` flag.

Figure 17: Successfully gained root.txt

7 Conclusion

This is my writeup for THM-LazyAdmin.
– Zackary Jordan